

Second Review Presentation

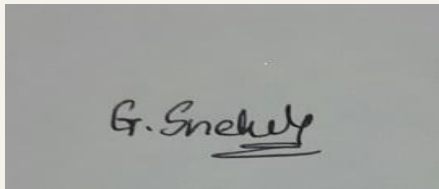
Cellular Automata and Algorithmic Preprocessing to Improve Clustering

Guide

Kamalika Bhattacharjee

Dr. Kamalika Bhattacharjee - Computer
Science and Engineering

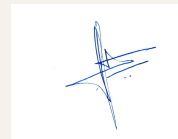
Team 8



Ganta Sneha Rao - 106119037

Subramanian

Subramanian V V - 106119123




Viswonathan Manoranjan - 106119145



Introduction

- Clustering is grouping of objects based on similarity such that the data in one group are more similar to each other than the other group.
- It is a common problem in unsupervised learning and there are multiple algorithms available for clustering.
- Clustering is crucial because it determines the intrinsic grouping of the unlabeled data present.

Problem Statement

- A non-conventional approach towards clustering, by grouping similar data points based on the rule and cycle formation using cellular automata.
 - The goal is to use a suitable encoding for the dataset and provide an efficient algorithm for clustering the data using the available CA rules.
- 

Objectives

Determine the suitable hash functions for entities that need clustering to be positioned in different cells during the initial approach.

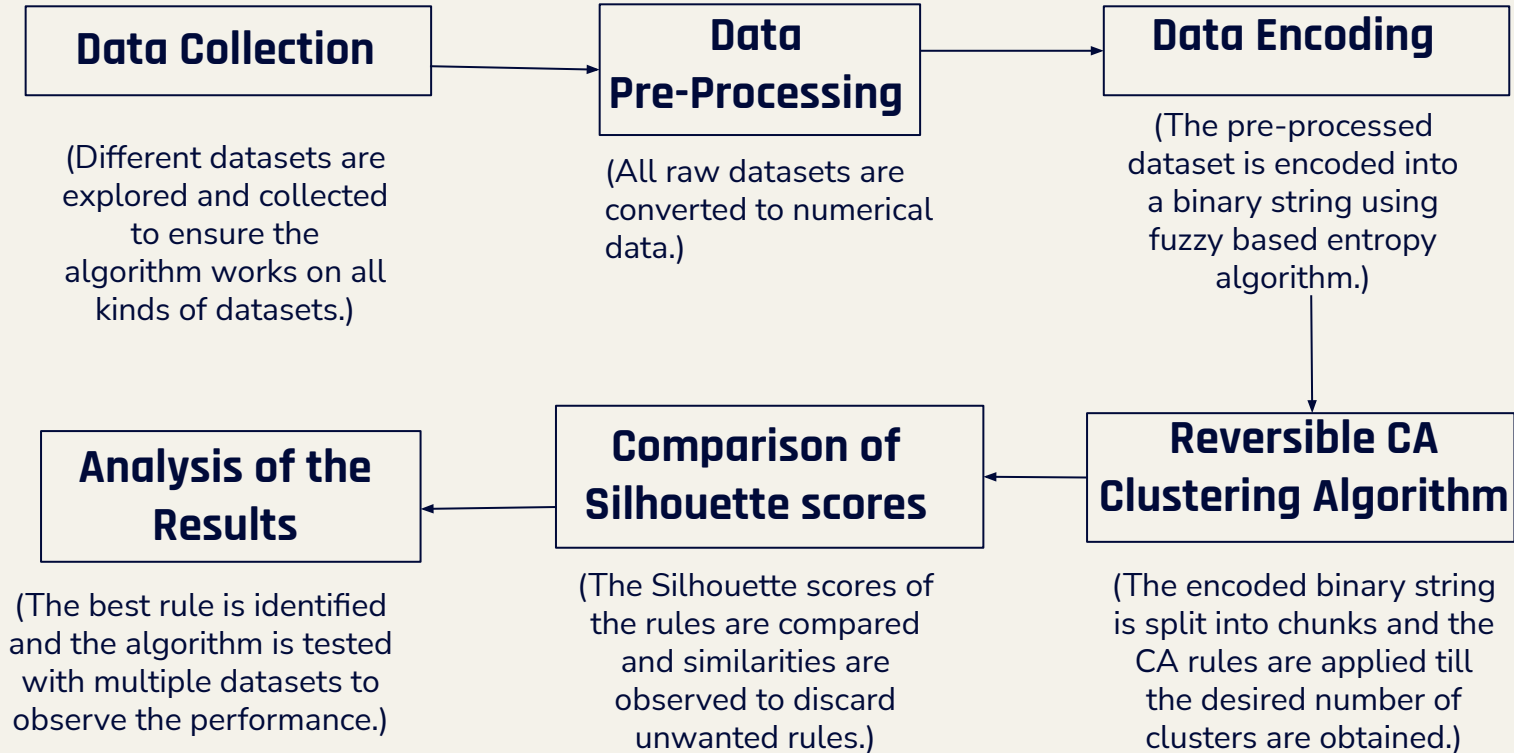
Propose a new algorithm that avoids the randomized search for rules for a particular dataset.

Package the whole process for ease of replication and development.

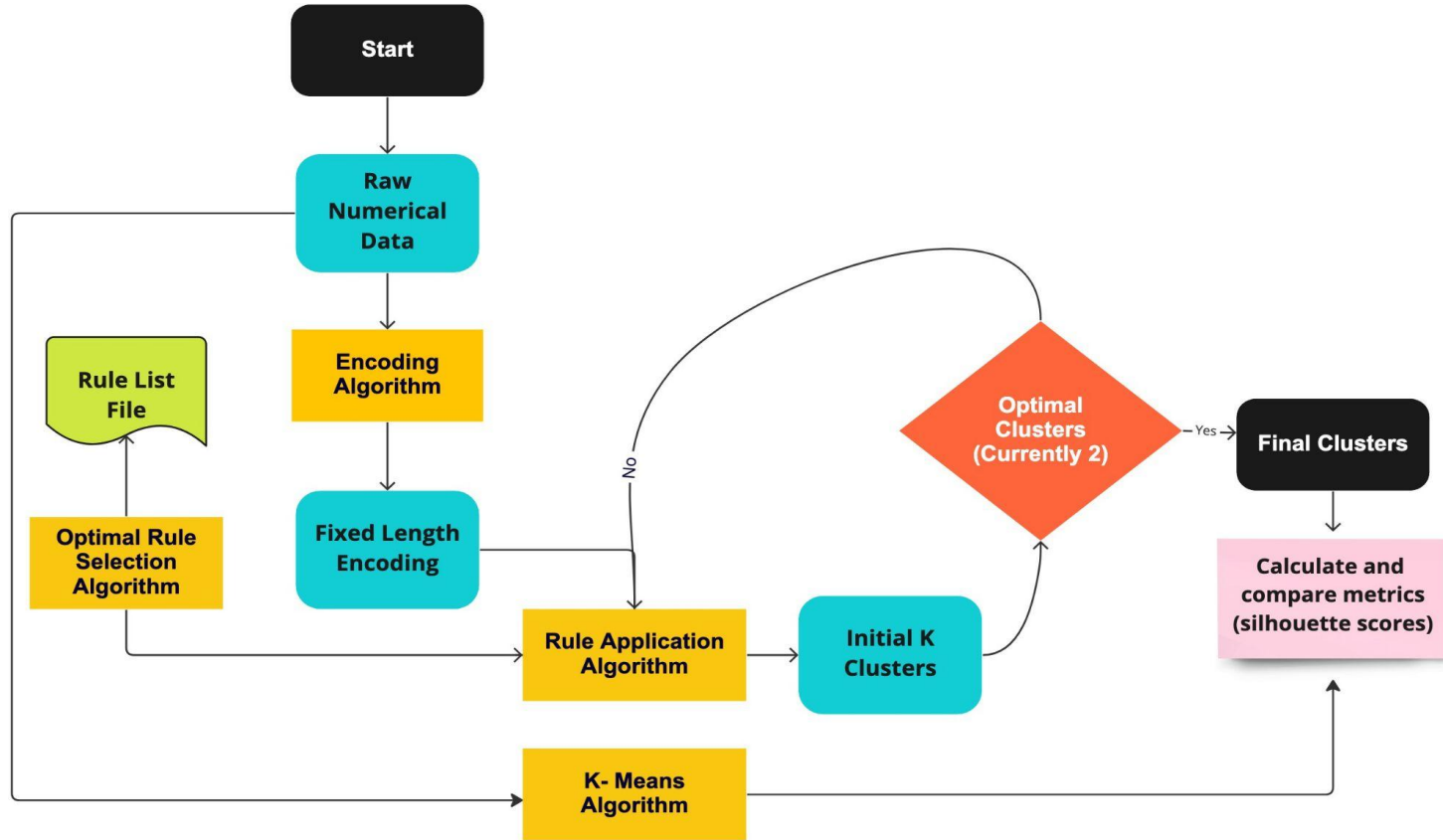
Provide flexibility for multiple datasets and variable window sizes.



Block Diagram



Workflow Diagram





Algorithm of Completed Modules

Stage 1 – Vertical Splitting:

Here, each encoded binary string of size n is split into some divisions. Each split should have even number of bits such that the split size is from (6, 14).

Stage 2 - Rule Selection:

For each of the splits, we find all possible combinations of the CA rules from the 162 candidate CAs which are reversible for the given split size and run them in parallel. These CAs distribute the partitioned configurations into some preliminary cycles. The (partitioned) configurations under the same cycle form a unique cluster.

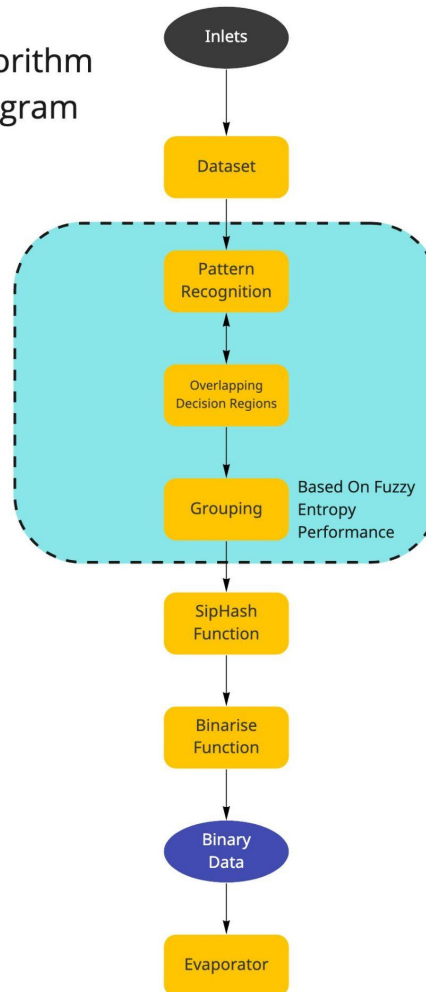
Stage 3: First Level Clustering:

Use output of Stage 1 and cluster further by concatenating the encoded strings of each split. Let the new length of configurations is n' . Our ideal target is to make this $n' \leq 12$, if it is larger than this, we repeat Stage 1 again on the new encoded strings. Apply a new rule to the new strings (modified configurations) until all are clustered.

Stage 4 - Getting Desired Number Of Clusters:

We obtain desired number of clusters by merging the clusters till desired number is reached. We iterate through the clusters to find one with the least number of elements (smallest cluster). The elements in the smallest cluster are extracted from that cluster and added to other clusters. The best fit for each element is determined by the clusters which give the highest performance index scores. This procedure is followed till we reach desired number.

Encoding Algorithm Workflow Diagram





Algorithm of Completed Modules

Encoding Stage

Step 1: Use **Fuzzy Based Entropy** algorithm to reduce the dimensionality of the data and also will discard all the unimportant, redundant, and noise-corrupted features.

Step 2: Evaluating information of the distribution of pattern within the pattern space.

Step 3: The pattern space may be partitioned into the **decision regions** that are **overlapping** in terms of complexity as well as load of computation for the classifier that may be reduced.

Step 4: The decision regions will be grouped based on the good classification performance as they are determined rightly using the fuzzy entropy measure that has been proposed. The reduced dataset is obtained.

Step 5: Use the in-built **hash()** Python function to encode the data features and convert it to binary strings. The in-built **hash()** function uses **SipHash**, which computes a **64-bit message authentication code** from a variable-length message and **128-bit secret key**.

Step 6: Concatenate the binary strings to get the binary string of length **63**.

Partial Results

Encoding Scheme : Fuzzy Entropy

Number of Trials : 1500

	Trial_no	Rule 1	Rule 2	Initial clusters	CA Silhoutte	Kmeans Silhoutte	New Clusters	New CA Silhoutte	New Kmeans Silhoutte
0	0	264499440	3027809415	2	0.671680	0.766361	2	0.671680	0.766361
1	11	2526451350	265482450	24	-0.033092	0.222334	2	0.605466	0.766361
2	97	1016918172	3789677025	27	0.104200	0.174086	2	0.577628	0.766361
3	1	267448335	1799965515	5	-0.204544	0.634825	2	0.573259	0.766361
4	89	2777130375	1267157835	3	0.506391	0.520635	2	0.570908	0.766361
5	284	3035673615	267448335	27	0.104523	0.174086	2	0.570619	0.766361
6	126	2783028705	267448560	27	0.094178	0.174086	2	0.570619	0.766361
7	383	3063191190	267390795	5	-0.361726	0.634825	2	0.559835	0.766361
8	4	267422991	267448560	28	0.156250	0.156250	2	0.540689	0.766361
9	10	267422991	755969295	28	0.156250	0.156250	2	0.540689	0.766361
10	3	267422991	267448335	28	0.156250	0.156250	2	0.540689	0.766361

Baseline Results

Encoding Scheme : Frequency Based Encoding

Number of Trials : 1500

	Trial_no	Rule 1	Rule 2	Initial clusters	CA Silhoutte	Kmeans Silhoutte	New Clusters	New CA Silhoutte	New Kmeans Silhoutte
0	301	3035673615	1259293560	3	0.673609	0.520635	2	0.746993	0.766361
1	107	2783028705	4027544304	16	-0.160179	0.409430	2	0.709970	0.766361
2	104	2783028705	3785744805	11	-0.389642	0.485542	2	0.708771	0.766361
3	20	1263225675	3530698098	27	0.136809	0.174086	2	0.605554	0.766361
4	113	2018212020	2273806215	25	0.033248	0.212747	2	0.605429	0.766361
5	361	3063191190	3724750095	3	0.545511	0.520635	2	0.594055	0.766361
6	114	252645135	3537972705	3	0.220068	0.520635	2	0.583336	0.766361
7	42	517136850	4034007024	27	0.105456	0.174086	2	0.570619	0.766361
8	10	255594255	255652080	26	0.097123	0.203847	2	0.570619	0.766361
9	16	517136850	2463552210	27	0.113996	0.174086	2	0.570619	0.766361
10	229	3031741620	265482450	26	0.067117	0.203847	2	0.553803	0.766361

Implementation/ Simulation Environment

Environment

- ❖ Python == 3.11.0
 - ❖ Scikit-learn == 1.2.0
 - ❖ Pandas == 1.5.2
 - ❖ Numpy == 1.23.5
-
- Processor : Intel i7 11700K 16M Cache, up to 5.00 GH
 - Graphics Card : NVIDIA GeForce RTX 3060
 - Disk : 1TB SSD

01

Python

For ease of understanding, functional programming, GPU compatibility and replicability of previous papers

02

Numpy and Pandas

For data handling and faster/parallel computations

03

Sklearn

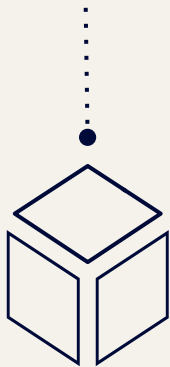
Employing K-Means clustering for baseline scores and calculating silhouette, calinski, davies boundary scores

Timeline

Phase 1

Week 1-3

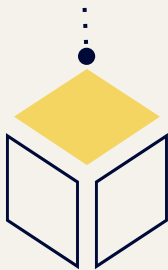
Go through the previous papers and replicate the results.



Phase 2

Week 4-8

Explore various patterns and develop a suitable encoding to cluster the data efficiently without much loss.



Phase 3

Week 9-13

Implement a suitable rule-selection algorithm for a dataset to improve performance.



Phase 4

Week 14-15

Record the results and package the whole process for easy usage.



References

- [1] Sukanya Mukherjee, Kamalika Bhattacharjee, and Sukanta Das. 2020. **Cycle Based Clustering Using Reversible Cellular Automata**. In *Cellular Automata and Discrete Complex Systems: 26th IFIP WG 1.5 International Workshop, AUTOMATA 2020, Stockholm, Sweden, August 10–12, 2020, Proceedings*. Springer-Verlag, Berlin, Heidelberg, 29–42.
- [2] S. Mukherjee, K. Bhattacharjee and S. Das, “**Clustering Using Cyclic Spaces of Reversible Cellular Automata**,” *Complex Systems*, 30(2), 2021 pp. 205–237.
- [3] Hong He, Yonghong Tan, “**Automatic pattern recognition of ECG signals using entropy-based adaptive dimensionality reduction and clustering**”, *Applied Soft Computing*, Volume 55, 2017
- [4] Mukherjee, Sukanya & Bhattacharjee, Kamalika & Das, Sukanta. (2021). “**Reversible Cellular Automata: A Natural Clustering Technique**. *Journal of Cellular Automata*”. 16. 1-38.
- [5] Abhishek S, Mohammed Dharwish, Amit Das, and Kamalika Bhattacharjee, “**A Cellular Automata Based Clustering Technique for High-Dimensional Data**”, *Proceedings of the Second Asian Symposium on Cellular Automata Technology 2023, IEST Shibpur*